Trajectory Generation and Path Planning for Autonomous Aerobots

S. Sharma
RASCAL Laboratory
The University of California Davis
Bainer Hall 1030B
Davis, California 95616
Email: sharma@ucdavis.edu

E.A. Kulczycki, A. Elfes
Robotics and Autonomy Section
The Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 911909
Email: Eric.A.Kulczycki@jpl.nasa.gov

Abstract—**This paper presents global path planning algorithms for the Titan aerobot based on user defined waypoints in 2D and 3D space. The algorithms were implemented using information obtained through a planner user interface. The trajectory planning algorithms were designed to accurately represent the aerobot's characteristics, such as minimum turning radius. Additionally, trajectory planning techniques were implemented to allow for surveying of a planar area based solely on camera fields of view, airship altitude, and the location of the planar area's perimeter. The developed paths allow for planar navigation and three-dimensional path planning. These calculated trajectories are optimized to produce the shortest possible path while still remaining within realistic bounds of airship dynamics.**

I. INTRODUCTION

Path planning techniques have become frequently used components in autonomous planetary exploration, in which trajectories are generated solely on vision-based data. The optimal path of an autonomous vehicle has a higher degree of accuracy when integrating specific vehicle dynamics, which has been performed for various robotic vehicles and aircraft. To perform the desired analysis of planetary bodies, such as Titan, one of Saturn's moons, an autonomous vehicle must be created capable of exploring the planet and collecting data; the vehicle must act as a scientist in which its observations will be determine an appropriate course of navigation. An aerobot, seen in Fig. 1, represents one such vehicle and is under current development at the Jet Propulsion Laboratory. This Titan Aerobot must have the ability to autonomously create an appropriate trajectory utilizing path planning techniques. A waypoint navigation system for the aerobot was developed to demonstrate initial path planning solutions in both planar and 3D space.

This system utilizes a simple graphical user interface and trajectory planning techniques to determine the optimal path of the airship based on given waypoints. This navigation system was created to allow the user to indicate a series of points or overall area desired for exploration. This information is then used in the navigation system to illustrate the desired path of the airship. The computed path allows for a trajectory to be developed that accounts for the overall goal of exploration and also integrates the dynamics of the airship. The navigation system allows the user to indicate a variety of factors, which affect the overall path the airship will ravel. These factors include the allowable turn radius of the vehicle and the amount of area covered by a camera mounted on the vehicle.

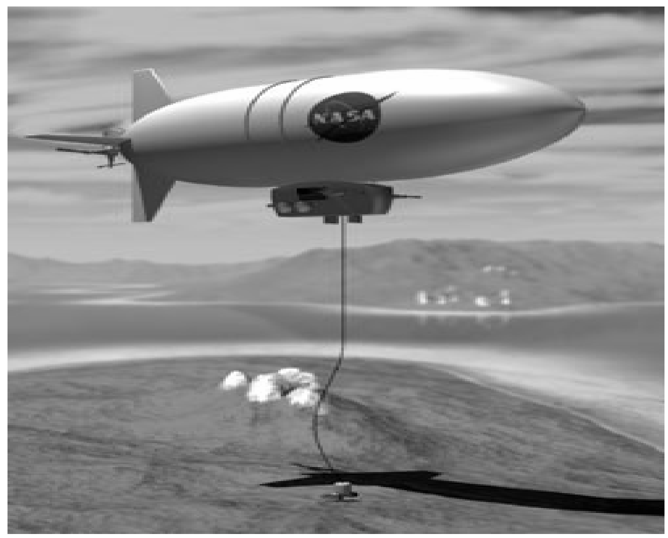


Fig. 1. An artist rendering of the Titan Aerobot performing scientific experiments on the moon's surface.

Several advanced algorithms capable of producing planar trajectories and three-dimensional paths exist, many of which are optimized along shortest distance parameters. This path planning algorithm incorporates several methods of waypoint identification with trajectory planning techniques while also including specific airship dynamics. Additionally, the ability to map a planar area using only a small amount of information utilizes path planning techniques in a novel manner. Constraints on the current system include the assumption that the aerobot is moving at a constant speed, eliminating time as a variable in trajectory planning. An additional constraint is that trajectory planning up to this point is independent of altitude. The airship is assumed to remain at a constant altitude when stationary due to the fact that there are relatively low winds on Titan. Furthermore assuming the airship is balanced, the aerobot should stabilize at a given altitude under small perturbations.

II. Related Work

Global path planning and dynamic path planning techniques have been explored extensively for robot navigation on planar surfaces and UAV navigation. Current research involves obstacle avoidance and incorporating vision based information into optimal path planning techniques. Popular algorithms serve to break up the observed environment into several regions from which a path can be interpolated in a piece wise fashion. In order to incorporate the dynamic behavior of an aircraft the completed path undergoes a smoothing technique that matches the plane's turning behavior. The 3D path planning algorithm presented in this paper interpolates a spline between given waypoints, in which the spline's behavior incorporates the dynamics of the aerobot producing a more accurate trajectory.

Richards and How [5] illustrate optimal path planning for an aircraft in which the dynamics of the vehicle are incorporated through limited turn rate. This global path planning ability is mated with dynamic collision avoidance algorithms. The path is generated at discrete intervals through a set of waypoints and is optimized for minimum completion time. This paper proposes the most integrated trajectory planning and vehicle dynamic approach however it is primarily concerned with shortest time path planning. Additionally, the path planning is calculated for only waypoints in two dimensions.

Navigation based on forming mosaic images autonomously generally lies in the area of underwater vehicle research. Traditional mosaic building has relied heavily on GPS and satellite information, however in underwater regions where such tools are unable to operate effectively new methods capable of mapping the sea floor had advanced mosaic based navigation. Gracia et al [6] explore building high quality mosaics of the sea floor including topography information. Their technique produces spatially coherent mosaics formed using a loop trajectory between a starting and final location in which the shortest path is sought while remaining within the mosaic border. The algorithm produces zig zag scanning motion producing a pattern that lacks consistency and fails to ensure coverage of a given area.

III. Trajectory Generation Methods

The goal of the path planning system is to allow a user to quickly obtain an approximate path of the airship given a set of waypoints. The path planner was first produced to project a trajectory for a set of points at a constant altitude producing planar navigation. This path planning was then expanded into a third dimension as altitude was incorporated into the trajectory planning scheme. The third method explored involved producing an appropriate trajectory to ensure complete coverage of a given area based on a camera's field of view. In order to complete the planar and 3D path planning it was first necessary to devise an interface that would allow the user to easily identify a series of points to which the trajectory can be applied.

A. Planner User Interface

The architecture of the interface, which can be seen in Fig. 2, was developed to allow the user to designate the waypoints using a variety of simple functions. The first function requires the user to input coordinates directly into the main figure of the interface thus allowing for quick and easy waypoint identification in three dimensions. The second method of waypoint identification allows the user to load a data file containing vectors with the waypoint coordinates. The data file allows an infinite set of waypoints to be entered in either 2D or 3D.

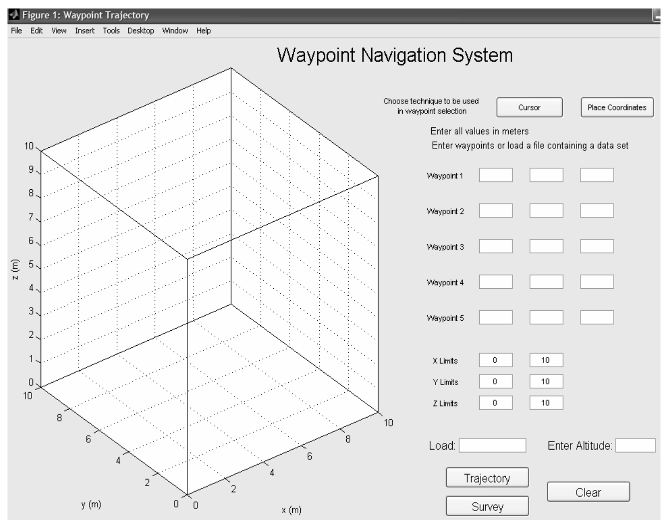


Fig. 2. The graphical user interface developed to allow for simple waypoint identification for trajectory planning methods.

The third function introduced in the interface allows the user to select various waypoints at a constant altitude using the cursor. Upon selection of the cursor button a set of 2D axes appears allowing the user to indicate desired points on this grid. After specifying the constant altitude of the airship for the given points the coordinates are then stored and shown in three dimensions. The interface was created using MATLAB to permit integration of the waypoints into the navigation system of a SIMULINK model of the aerobot. How are these point available for trajectory planning.

B. Path Planning Heuristics

The approach taken for aerobot trajectory planning involves optimizing the path for the shortest Euclidean distance. This linear path then provides the necessary information for the spline to produce the most cost effective path that weights the distance between waypoints and the possible turning radius of the aerobot. Essentially the current waypoint is used to determine the estimate cost to the destination based on the shortest distance between the points. By optimizing this initial feasible path the generated path produces the most accurate airship trajectory. To determine the distance between the linear interpolation and the spline trajectory the points that provide the least distance for the given interval is found. The closest two points are found after several iterations and are then evaluated. The distance between the two points is found giving an approximation of the minimum distance between the two segments. This allows the minimum turning distance requirement to be constantly determined and satisfied. The grid upon which the waypoints are mapped is given as a fine mesh in three dimensions, allowing for the interpolated path to smoothly represent the trajectory.

The path planning analyzes the Euclidean distance between successive waypoints to determine if such a trajectory is feasible for the airship. If the predicted path between two waypoints cannot be realistically completed given the dynamics of the airship, the point is ignored and the trajectory between the current waypoint and successive point is considered.

C. *Planar Trajectory Navigation*

The first trajectory method involves creating a linear path at a constant altitude from waypoint to waypoint in a piecewise manner. This Euclidean path was determined utilizing the linear interpolation polynomial,

$$g(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \qquad (1)$$

for every successive pair of waypoints. Where the variable x represents the x coordinate and f(x) is given to be the y coordinate. Upon determination of the linear path additional waypoints are placed incrementally along the trajectory. The distance between the two waypoints in question determines the placement and quantity of any additional waypoints. These generated waypoints will serve as markers upon which the aerobot can correct its trajectory if necessary. Furthermore, the additional waypoints allow for the minimum turning radius of the aerobot to be incorporated into the spline trajectory once it has been found.

Utilizing the popular Thomas Algorithm, a tridiagonal system is solved in order to produce a planar cubic spline between waypoints. This piecewise cubic function serves to smooth the trajectory and incorporates the airship dynamic constraints by first solving for the distance between the linear path and cubic spline. This information is then used to calculate the smallest radius for each sub-path, which is then compared to the give constraint. To ensure the constraint is met the slopes of the cubic spline sub-path is modified using experimental results. This procedure continues for every sub-path given by the set of waypoints until an accurate cubic spline trajectory has been produced.

D. *3D Trajectory Navigation*

The trajectory methods applied in 2D are adapted and are similarly used in three-dimensional trajectory planning. The first step in formulating the 3D path involves determining the linear piecewise interpolation for each sub-path. The linear interpolation is completed using parameterization of the path between a given waypoint set. Additional waypoints are introduced along a given sub-path in the method described in forming the planar trajectory. This linear path is then utilized to determine an appropriate spline trajectory.

A piecewise Bezier interpolation was utilized to produce an appropriate path for the airship due to the fact that the properties of the curve, such as tension and continuity, can be manipulated through a simple matrix. The curve is defined geometrically and produces a resultant parametric function to represent the path. The cubic Bezier curve is characterized by the following: the curve passes through the beginning and end control points for each sub-path, the curve is continuous and has continuous derivatives, and second degree Bernstein polynomials are used to blend the control points. The cubic Bezier curve is found utilizing four control points and can be defined mathematically using the following,

$$\left.\begin{aligned}
F(t) &= \sum_{k=0}^{3} P_k \frac{3!}{k!(3-k)!} t^k (1-t)^{3-k} \\
F(t) &= P_0(1-t)^3 + 3(P_1 t(1-t)^2 + 3P_2 t^2(1-t) + P_3 t^3 \\
F(t) &= t^3(-P_0+3P_1-3P_2+P_3) + t^2(3P_0-6P_1+3P_2) + t(-3P_0+P_1) + P_0 \\
F(t) &= t^3(P_3+3(P_1+P_2)-P_0) + 3t^2(P_0-2P_1+P_2) + 3t(P_1-P_0) + P_0
\end{aligned}\right\} 0 \le t \le 1 \qquad (2)$$

where P_0, P_1, P_2, P_3, and P_4 are the control points for a given curve and t represents the parameterization used to produce the function, F(t).

The cubic Bezier function is optimized with regard to distance and is completed using the linear interpolation trajectory. Furthermore, the tension and continuity parameters are iteratively evaluated for every sub-path to ensure the minimum turning radius constraint is met in generating the trajectory.

E. *Planar Area Vision Mapping*

Basic algorithms capable of producing mosaic mapping exists, this field is expanded upon through the creation of a trajectory that ensures complete coverage of a given area using only a small amount of information. Initial assumptions hold the airship at a constant speed and constant altitude during such a survey mission. The necessary information required for planar area mapping include the field of view of the camera, the altitude of the aerobot, and the perimeter location of the area desired to be mapped.

Utilizing the camera field of view and the altitude of the airship the area of the obtainable image can be found. The area of this image is represented by a rectangular outline which then is placed in an overlapping pattern to cover the given planar region. The amount of overlap of the images has been set at thirty five percent to allow for optimal mosaic creation, however this value can be easily altered based on desired specifications. The trajectory initiates at a set of coordinates based on defined perimeter values, from which a survey pattern commences along a boundary of the given area. Subsequent rows are formed using the given image area placed at an offset in order to remove the possibility of imageless areas when forming the mosaic.

The waypoints necessary to form the trajectory are obtained from the center of each rectangular image, thus ensuring complete coverage of the area. Based on the position of the rows within the perimeter additional waypoints are formulated to allow a smooth path as the airship traverses row to row. The planned path thus created can be found in both 2D and 3D with the appropriate methods described in the prior sections.

IV. Results

The trajectory generation methods presented above were used to produce experimental results given in this section. The first result can be seen in Fig. 3. The planar path planning method is utilized to produce the trajectory illustrating both the linear interpolation and the cubic polynomial altered to account for the minimum turning radius of the airship. The planar path only deviates from the Euclidean path to ensure the dynamics of the airship can be maintained, producing an distance optimal trajectory. The algorithm generates additional waypoints along the linear path from which the spline trajectory is modified at many intervals along each sub-path. Additionally, the original set of waypoints involved a pair of eight coordinates as can be seen in Fig. 3. Due to the close proximity of the waypoint it was essentially passed over and a sub-path was created utilizing the successive coordinate pair. A smoothing algorithm was not necessary for the spline, but can be generally applied using a general Gaussian algorithm.

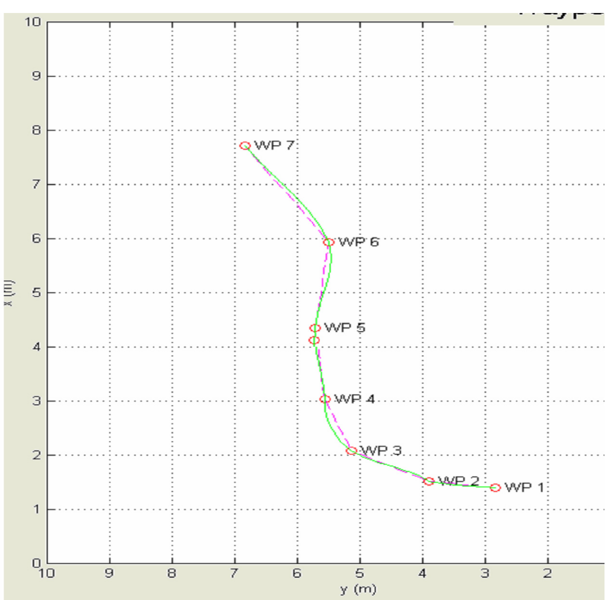


Fig. 3. The trajectory of the airship given a set of eight waypoints at a constant altitude, illustrating the planar path planning method.

The 3D path planning result is given in Fig. 4. Using a set of given three-dimensional coordinates a hypercube mesh was applied to the overall area with the waypoint position stored for use. The linear interpolation in 3D is then completed and additional waypoints along each sub-path are calculated, these can be seen as the green circles in the figure. The linear interpolation allows the distance of each sub-path to be found, this information is then used to determine the necessity of passing over a given waypoint due to its close proximity to other point in the data set.

The cubic Bezier interpolation is then performed based on tension and continuity values determined experimentally. Along each sub-path the center of turning is assumed to rest on the linear path, using this information the turning radius at every point along the spline is found. The allowable turning radius constraint is thus continuously checked along every sub-path and if necessary the tension and continuity matrix controlling the Bezier spline is altered in response. The four control points of the cubic Bezier can also be altered in order to ensure the aerobot has the ability to turn at the defined waypoints. The curvature of the spline at the user-defined waypoints is utilized to determine if this constraint is met.

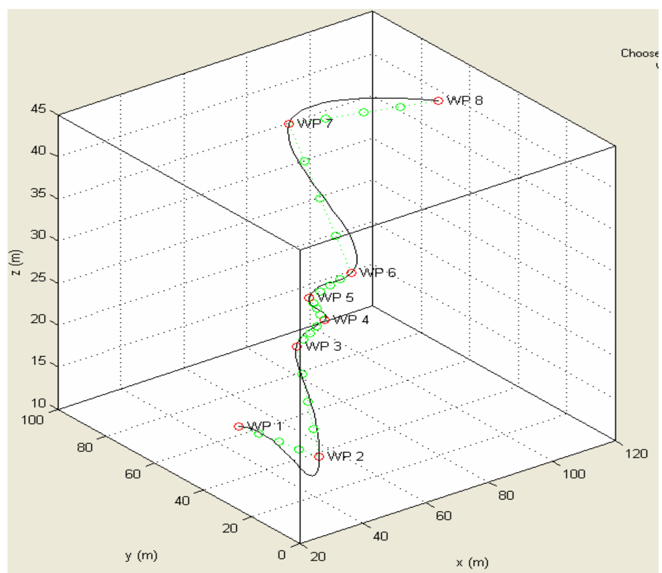
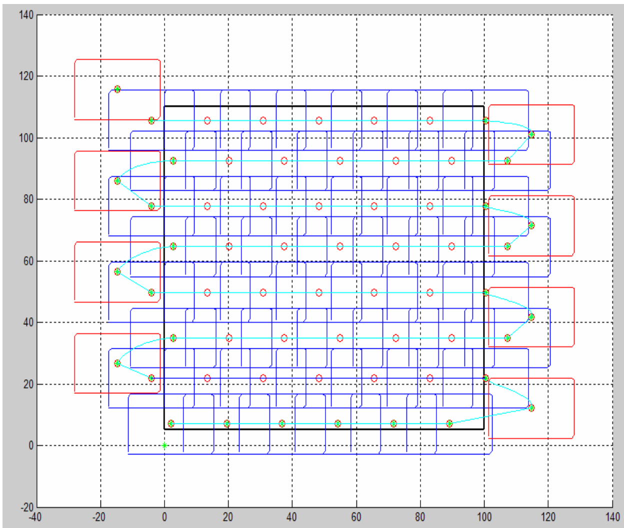


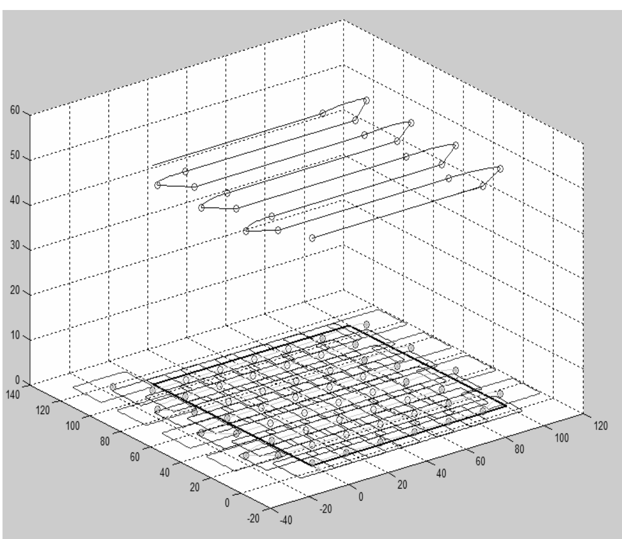
Fig. 4. The 3D trajectory of the aerobot is shown utilizing a linear interpolation and cubic Bezier method.

The final path planning result is given in Fig. 5, in which the planar survey method can be seen. For this result a camera with a horizontal field of view of 30 degrees and a vertical field of view of 20 degrees was used. The aerobot's altitude was set at 60 meters and perimeter desired to be mapped was given as a set of x and y limits. The trajectory commences at a point that is determined using the defined perimeter values and the altitude of the airship.

The camera fields of view are then used to produce the rectangular regions seen in the figure. The number of rectangular images required to cover the given area is then determined and the trajectory then begins its course. An overlap set at 35% was utilized to produce this result. This requires that 35% of the length and width of the rectangular region overlap with another rectangular image. The vision areas are then appropriately overlapped throughout the perimeter of the prescribed area. Subsequent rows are set at an offset to guarantee complete coverage. The centers of the rectangular regions are then stored as the waypoints for the planar mapping trajectory.



(a)



(b)

Fig. 5. The planar survey mapping trajectory of the aerobot utilizing a camera field of view, defined perimeter, and the altitude of the vehicle. (a) The area is first mapped using the rectangular regions that will be captured by the camera mounted on the aerobot. (b) The waypoints from the center of each vision area is then used to create an appropriate trajectory that will allow the entire area to be mapped to form a mosaic.

Additional waypoints along the perimeter of the area ensure the airship correctly turns from one row to another in order to produce the desired planar mapping trajectory. The final trajectory of the aerobot is seen in Fig. 5 (b) in which the planar trajectory is applied to vision mapping waypoints.

V. SUMMARY

This paper introduced path planning methods in 2D and 3D for autonomous aerobots that incorporates vehicle dynamics and allows for a unique area mapping function. A simple user interface was created to allow for several methods of waypoint identification from which a trajectory may be generated. The trajectory path utilizes linear interpolation methods, cubic splines, and piecewise Bezier curves to approximate the path of an aerobot. Incorporating the dynamics of the aerobot allows a more accurate trajectory to be computed. The planned path is currently optimized in terms of distance and incorporated a yawing constraint. The trajectory can be optimized with regard to time and additional constraints such as in pitch can also be included. This trajectory can further be incorporated into the navigation system of an aerobot simulation model, this will allow a planned trajectory to be compared with that of the aerobot under various flight conditions.

The trajectory planning methods can be utilized in UAV research and can also be of use in underwater vehicle systems. The vision area mapping trajectory can be used extensively in the UAV realm in order to continuously monitor a certain region, such as a no fly zone. Vision capabilities will replace user defined waypoints thus allowing autonomous vehicles to follow the generated trajectory based on mission specifications.

ACKNOWLEDGMENT

The research was made possible through contributions from Adnan Ansar, Hrand Aghazarian, and Dave Bayard. This work was supported by the Internal Research and Technology Development Program at the Jet Propulsion Laboratory under the Summer Undergraduate Research Fellowship program.

REFERENCES

[1] S.B.V. Gomes, "An Investigation of the Flight Dynamics of Airships with Application to the YEZ-2A," Ph.D Thesis, College of Aeronautics, Cranfield University, October 1990

[2] S.B.V. Gomes, J.J.G. Ramos, "Airship Dynamic Modeling for Autonomous Operation," IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998

[3] J.L. Hall, V.V. Kerzhanovich, A.H. Yayrauian, J.A. Jones, C.V. White, B.A. Dudik, G.A. Plett, J. Mennella, A. Elfes, "An Aerobot for Global *In Situ* Exploration of Titan," COSPAR04-A-00198, 35th COSPAR Scientific Assembly, Paris

[4] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1988.

[1] A. Richards, J. How, Aircraft trajectory planning with collision avoidance using mixed integer linear programming, in: IEEE American Control Conference, May, 2002.

[2] N.R. Gracias, S. van der Zwaan, A. Bernardino, J. Santos-Victor, "Mosaic-based navigation for autonomous underwater vehicles," *Oceanic Engineering, IEEE Journal of* , vol.28, no.4pp. 609- 624, Oct. 2003

[5] J. Kuffner, "Effective Sampling and Distance Metrics for 3D Rigid Body Path Planning," *Proc. 2004 IEEE Int'l Conf. on Robotics and Automation (ICRA 2004)*, IEEE, May, 2004.

[6] N. Gracias and J. Santos-Victor, "Trajectory reconstruction with uncertainty estimation using mosaic registration," *Robot. Autonomous Syst.*, vol. 35, pp. 163-177, July 2001.

[7] J.J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* , vol.4, no.pp. 3993- 3998 Vol.4, April 26-May 1, 2004

[8] D. Ferguson, M. Likhachev, and A. Stentz, "A Guide to Heuristic-based Path Planning," Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS), June, 2005.